

MainActivity (1)

```
package com.example.weather
import ...

class MainActivity : AppCompatActivity() {
    var api_key = "374344ab9934b08326458b7158b8dc9e"
    private lateinit var btVar1: Button
    private lateinit var textView: TextView
    //FusedLocationProviderClient – это высокоуровневый API для отслеживания местоположения, предоставляемый Google Play Services.
    private lateinit var fusedLocationClient: FusedLocationProviderClient
    // Код запроса разрешения на доступ к местоположению
    private val LOCATION_PERMISSION_REQUEST_CODE = 1
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
            insets
        }

        textView = findViewById(R.id.text)
        btVar1 = findViewById(R.id.button)
        // Создаем экземпляр клиента для получения местоположения
        fusedLocationClient =
            LocationServices.getFusedLocationProviderClient( activity: this)
        // Проверяем разрешения на доступ к местоположению
        btVar1.setOnClickListener {
            textView.text = " "
            checkForPermission()
        }
    }
    // Метод для проверки наличия разрешений
    private fun checkForPermission() {
        if (ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_FINE_LOCATION)
            != PackageManager.PERMISSION_GRANTED
            && ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_COARSE_LOCATION)
            != PackageManager.PERMISSION_GRANTED) {
            // Запрашиваем разрешения, если они не предоставлены
            ActivityCompat.requestPermissions( activity: this,
                arrayOf(
                    Manifest.permission.ACCESS_FINE_LOCATION,
                    Manifest.permission.ACCESS_COARSE_LOCATION ),
                    LOCATION_PERMISSION_REQUEST_CODE)
        } else {
            // Если разрешения уже предоставлены, получаем местоположение
            obtainLocation()
        }
    }
}
```

MainActivity (2)

```
// Метод, который обрабатывает результат запроса разрешений
override fun onRequestPermissionsResult(
    requestCode: Int, permissions:
    Array<out String>, grantResults: IntArray) {
    super.onRequestPermissionsResult(
        requestCode, permissions,
        grantResults
    )
    // Проверяем код запроса разрешений
    if (requestCode == LOCATION_PERMISSION_REQUEST_CODE) {

        // Если разрешение было предоставлено
        if ((grantResults.isNotEmpty() && grantResults[0] == PackageManager.PERMISSION_GRANTED)) {
            // Получаем местоположение
            obtainLocation()
        } else {
            // Если разрешение было отклонено, показываем сообщение
            Toast.makeText( context: this, text: "Разрешение отклонено", Toast.LENGTH_SHORT).show()
        }
    }
}

@SuppressLint("MissingPermission")
private fun obtainLocation() {
    // Получаем последнее известное местоположение
    fusedLocationClient.lastLocation.addOnSuccessListener { location: Location? ->
        // Проверяем, что местоположение не равно null
        if (location != null) {
            Log.e( tag: "LOCATION", location.latitude.toString())
            // Формируем URL для запроса к API погоды с текущими координатами
            val weatherUrl = "https://api.openweathermap.org/data/2.5/weather?lat=" +
                "${location.latitude}&lon=${ location.longitude }&units = metric&appid = ${ api_key }"
            Log.e( tag: "URL", weatherUrl)
            // Запрашиваем температуру по текущему местоположению
            getTemp(weatherUrl)
        } else {
            // Если не удалось получить местоположение, выводим сообщение
            Toast.makeText( context: this, text: "Не удалось получить местоположение", Toast.LENGTH_SHORT).show()
        }
    }
}

//это метод, который используется в Android для обработки неудачных результатов асинхронных операций
.addOnFailureListener { exception ->
    // Если не удалось получить местоположение, показываем сообщение
    Toast.makeText( context: this, text: "Location Permission not granted", Toast.LENGTH_SHORT).show()
}
}
```

MainActivity (3)

```
private fun getTemp(url: String) {  
    // Создаем очередь для запросов  
    // Volley - это HTTP-библиотека, которая используется для  
    // кэширования и выполнения сетевого запроса в приложениях Android.  
    val queue = Volley.newRequestQueue(context: this)  
    // Выполняем запрос к API погоды  
    val stringReq = StringRequest(  
        Request.Method.GET, url, { response ->  
            val obj = JSONObject(response) // Получаем JSON-объект из ответа  
            val main: JSONObject = obj.getJSONObject(name: "main")  
            val temperature = main.getString(name: "temp") // Температура  
            val city = obj.getString(name: "name") // Город  
            val humidity = main.getString(name: "humidity") // Влажность  
            textView.text = "Город: $city \nТемпература: ${temperature}C \nВлажность: $humidity"  
        }, {  
            textView.text = "Ошибка!"  
        })  
    // Добавляем запрос в очередь  
    queue.add(stringReq)  
}
```

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#B2DFDB"
    android:orientation="vertical"
    android:gravity="center"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Город: $city \nТемпература: ${temperature}C \nВлажность: $humidity"
        android:textAlignment="center"
        android:textSize="25sp"
        android:textStyle="bold" />
    <Button
        android:id="@+id/button"
        android:layout_width="250dp"
        android:layout_height="55dp"
        android:layout_marginTop="150dp"
        android:textSize="20sp"
        android:text="Узнать погоду"
        app:cornerRadius="13dp" />
</LinearLayout>
```

Результат:

